

**AMENDMENTS TO THE CLAIMS**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

1. (Currently amended) A computer-implemented method for facilitating long-running transactions of a business workflow process, comprising:
  - reducing a business process to a scheduling programming language written in XML;
  - dividing the reduced business process into at least one independent transaction and at least one parent interdependent transaction, the at least one parent interdependent transaction comprises two or more non-uniform child interdependent transactions;
  - executing the at least one independent transaction independently from the at least one parent interdependent transaction to increase throughput and decrease latency of the business process, the at least one independent transaction commits upon successful execution;
  - executing the child interdependent transactions independently from each other, the at least one parent interdependent transaction commits when a last child interdependent transaction commits; and
  - transferring committed data associated with the at least one independent transaction and the at least one parent interdependent transaction to a computer component for further processing.
2. (Previously presented) The method of claim 1, the child interdependent transactions respectively include one or more actions, the one or more actions are concurrently executed independently from each other.
3. (Previously presented) The method of claim 2, respective child interdependent transactions commit when all of their associated actions are completed.

4. (Previously presented) The method of claim 1, further comprising explicitly defining transaction boundaries for the at least one independent transaction and the child interdependent transactions as a function of a number of actions within the at least one independent transaction and the child interdependent transactions, respectively, in order to define a granularity at an action level.

5. (Previously presented) The method of claim 1, the child interdependent transactions are concurrently executed in isolation from each other.

6. (Previously presented) The method of claim 1, further comprising employing separate machines to execute the at least one independent transaction and the at least one parent interdependent transaction.

7. (Currently amended) A system that uses a scheduling programming language written in XML for facilitating implementation of business processes within a computer-readable medium, the components comprising:

a user interface component; and

a plurality of model components accessible through the user interface component, the plurality of model components allows a user to create a model of a business process and reduce the model *via* the scheduling programming language, the plurality of model components comprises a distinguishing model component that distinguishes between concurrent autonomous business operations and concurrent interdependent business operations, the concurrent interdependent business operations being non-identical.

8. (Previously presented) The system of claim 7, further comprising a transaction grouping component that groups business operations into concurrent interdependent transactions.

9. (Previously presented) The system of claim 8, the transaction grouping component provides synchronization of concurrent interdependent transactions based on completion of the concurrent interdependent transactions.

10. (Previously presented) The system of claim 7, further comprising an action grouping component that groups business operations into concurrent interdependent actions.
11. (Previously presented) The system of claim 10, the action grouping component provides synchronization of concurrent interdependent actions based on completion of the concurrent interdependent actions.
12. (Previously presented) The system of claim 7, the plurality of components further comprising at least one boundary establishing component that defines transaction boundaries.
13. (Previously presented) The system of claim 12, the at least one boundary establishing component includes a component that establishes concurrent operations.
14. (Previously presented) The system of claim 12, the at least one boundary establishing component includes a component for establishing sequential operations.
15. (Previously presented) The system of claim 12, further comprising a compensation component that compensates committed interdependent concurrent transactions and is invoked upon the occurrence of a failed interdependent concurrent transaction.
16. (Previously presented) The system of claim 15, the interdependent concurrent transactions are children transactions in a parent transaction, wherein the compensation component is invoked by the parent transaction.
17. (Previously presented) The system of claim 15, the compensation component calls compensation routines within the committed interdependent concurrent transactions.

18. (Previously presented) The system of claim 15, the compensation component calls compensation routines within the failed interdependent concurrent transaction.
19. (Previously presented) The system of claim 18, the compensation routines utilize information within the committed interdependent concurrent transactions.
20. (Previously presented) The system of claim 15, the compensation component calls compensation routines within the failed interdependent concurrent transaction based on information on the committed interdependent concurrent transactions stored within a database.
21. (Previously presented) The system of claim 13, the computer readable medium resides on a computer system.
22. (Previously presented) The system of claim 7, the plurality of components further comprising a component that defines concurrent synchronizing constraints as occurring upon completion of the autonomous operations.
- 23-27. (Cancelled).

28. (Currently amended) A system that uses a scheduling programming language for representing business comprising:

means for distinguishing between synchronization of autonomous concurrent operations from interdependent concurrent operations, the autonomous concurrent operations and the interdependent concurrent operations are represented in the scheduling programming language, the scheduling programming language written in XML, the interdependent concurrent operations comprise pluralistic interdependent concurrent operations;

means for expressing synchronization constraints on completion of autonomous concurrent operations; and

means for allowing association of transaction operations and groups of business operations.

29. (Previously presented) The method of claim 1, further comprising failing the at least one parent interdependent transaction when at least one of its child interdependent transactions does not commit, and compensating the at least one failed child transaction.

30. (Previously presented) The method of claim 29, the at least one parent interdependent transaction invokes a compensation routine within the at least one failed child transaction that compensates the at least one failed child transaction.

31. (Previously presented) The method of claim 1, further comprising compensating the at least one independent transaction when at least one of the child interdependent transactions does not commit.

32. (Previously presented) The method of claim 1, further comprising compensating the at least one parent interdependent transaction when it does not commit and all of its child interdependent transactions commit.

33. (Previously presented) The method of claim 32, the at least one parent interdependent transaction invokes its own compensation routine.